The EPEC Algorithm for Vision Guided Manipulation: Analysis and Validation

Matthew DiCicco, Max Bajracharya, Kevin Nickels, Paul Backes
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109, USA
818-393-5606
mdicicco@jpl.nasa.gov

Abstract— This paper describes the simulated performance and experimental validation of a computationally efficient algorithm for improving positioning accuracy of robot arms using low speed feedback from fixed stereo cameras. The algorithm, called End-Effector Position Error Compensation (EPEC) is robust to visual occlusion of the end-effector and does not require high fidelity calibration of either the arm or stereo camera. The algorithm works by calculating an error vector between the locations of a fiducial on the arm's end-effector as predicted by arm kinematics and detected by a stereo camera triangulation. With this knowledge, the commanded target pose is adjusted to compensate for positioning errors. A simulation environment where arbitrary error can be introduced into arm-camera systems is introduced and used to provide an assessment of the performance of the algorithm under both ideal and degraded conditions. Experimental results in the laboratory and on Mars are presented to validate the simulated performance estimates.

TABLE OF CONTENTS

1. INTRODUCTION

End-Effector Position Error Compensation (EPEC) is an algorithm for improving the positioning accuracy of robot manipulators in arm-camera systems like those employed on the Mars Exploration Rovers (MER). These rovers each have a 5 degree of freedom arm, called the Instrument Deployment Device (IDD), mounted to the rover base along with a pair of CCD cameras calibrated for stereo vision that are used to position science instruments on targets of interest.

Targets in this system are designated in stereo images by earth based controllers. The target designated through vision is triangulated into a 3-D point in the camera pair's coordinate system. When this position is uploaded to the arm, there is a visible discrepancy between the final position of the arm and the initial command. Currently, this discrepancy can be as much as 1cm [3]. This system requires three consecutive sols (planetary days) to complete. On sol 1, the command is sent to take images of the target. When the images are received, they are analyzed and a target is chosen. On sol 2, this target is uploaded along with a command to take another set of images to verify placement. On sol 3, the placement is verified in the second set of images and the science operation can commence [16].

The desire to increase science return from in-situ planetary exploration has driven a new trend of research to perform placement of an instrument on a distant (10m away) science target in one uplink command [12] [2]. In this task, a target is picked in the rover's long range cameras from as far as 10m away. The rover then autonomously drives toward the target, positions its base appropriately, and deploys its science instrument on the target. The allotted error budget for all of the combined steps in this activity is 1cm. As mentioned, the positioning error for the arm placement is on the order of 1cm. Algorithms for reducing this error are necessary to complete the single cycle instrument placement task. EPEC was developed to be conceptually simple, computationally efficient, robust to large sources of error, and degrade gracefully even when the final arm position is not in view of the cameras.

2. PRIOR AND RELATED WORK

This work has been performed for application to MER style arm/camera systems where a manipulator, typically 5 degrees of freedom, is rigidly mounted to the rover body along with a wide field of view (120°+) stereo pair of hazard avoidance cameras (hazcams) [10]. Science targets are designated in the hazcam images and the arm is used to place a science instrument on the target. Generally this is done using a purely kinematic model which includes a model of deflection due

1-4244-0525-4/07/$20.00/
IEEEAC paper # 1636

to gravity in the arm. On MER, this closed-loop system is accurate to approximately 1cm [3].

The state of the art for vision-based instrument placement on a planetary surface was the recent use of the Hybrid Image-Plane/Stereo (HIPS) algorithm [4] to update camera models offline for the MER rovers. HIPS is based on the Camera Space Manipulation (CSM) [14] [13] concept, but modified to work for very wide field of view cameras with a target close to the camera and perspective instead of orthographic camera models with a narrow baseline. However, because HIPS updates its camera model on each iteration (a nonlinear least-squares adjustment), the algorithm is quite computationally expensive. The version of HIPS used on the MER rovers used one set of offline calculations to generate new camera models. Online re-calculation during arm motion was not performed. The benefit of this approach is that it does not require well calibrated cameras and can handle occlusion of the end-effector fiducial.

Other related work includes Robonaut, a robotic astronaut's assistant being developed at Johnson Space Center, for which a hand-eye calibration method has been proposed for more accurate grasping [11]. This technique uses an end-effector mounted spherical target that is detected in the robot's head mounted stereo cameras to calibrate its 7DOF arm's kinematic model.

Visual servo control is well researched and has an established lexicon [7]. In this field, EPEC is a look-and-move system because the vision system is calculating commands which are fed to a joint space controller. It is a position based system because the error calculations are made in Cartesian space and not in camera space. Because both the target and end-effector are identified in the vision system, this is also an endpoint-closed loop system (ECL).

The major difference between EPEC and traditional visual servo systems is that EPEC generally requires only one iteration of the correction. This allows for either the use of newly calculated correction vectors or a lookup table to apply an appropriate correction vector, calculated earlier, without additional use of the stereo cameras. The primary advantage of this approach is the inherent safety. All commands from EPEC can be observed by ground based controllers before being uploaded to the spacecraft. The arm motion is never left solely to an onboard closed-loop visual servo system. This step is critical in in-situ rover operations.

The EPEC correction vectors can also be stored and observed as a diagnostic tool. In this way, EPEC is essentially an open-loop feed forward error compensator and an observer of arm/camera degradation over time.

After describing the details of the algorithm, these two ideas are addressed through simulation and experiment.

3. ALGORITHM DESCRIPTION

End-Effector Position Error Compensation (EPEC) works by taking the difference between the end-effector location calculated by kinematics and by vision, called the correction vector, and using it to modify the original target pose. This process is illustrated in Figure 1.

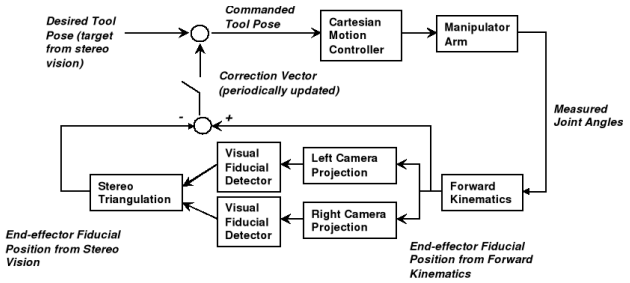

Figure 1. EPEC algorithm block diagram

The desired tool pose, P_{Tool}, is a vector of: x, y, z, azimuth, and elevation of the tool. It is designated in Cartesian space associated with the cameras, C_V. The arm kinematics operate in a second Cartesian space, manipulator space, designated C_M. It is assumed in this process that $C_V = C_M$. Inverse kinematics is performed on the target point to produce a set of joint angles corresponding to P_{Tool} in C_M, to which the arm is commanded to move. The process of taking images, identifying the fiducial, and triangulating the end-effector location identifies the tool location, X_{Tool} in C_V. At current, using one fiducial can only identify the location (x, y, z) and not the full pose (which includes azimuth and elevation angles). To accommodate this, P_{Tool} can be truncated in C_M by ignoring azimuth and elevation data to get X_{Tool}. The difference between the X_{Tool} value in C_V and C_M is the EPEC correction vector.

Basic operation of EPEC is to move to a target, calculate the correction vector, apply the correction vector to the target, and then move again with the corrected target. This process can be repeated in a servo loop indefinitely until the final error is below a desired threshold. However, with an operator checking each commanded move from EPEC, additional iterations will require additional sols of operation time. For the IDD scale manipulators investigated in this research, one iteration has generally been sufficient to achieve accuracy well below the 1cm baseline.

Additionally, because the correction vectors are intrinsically associated with the location in CV space where they are calculated, if a target is located close to a correction vector calculated previously it is possible to perform a feed-forward addition of this correction without performing the costly imagining and triangulation. If this system does not achieve the desired error tolerance, a full position based visual servo loop could be employed by iteratively taking more images and calculating and applying new correction vectors.

This algorithm is not limited to final target points. Intermediate waypoints could also be corrected for in the same manner by either adding the appropriate correction vector from a table or by stopping and calculating new vectors along a motion. The experimental and simulated robot/camera systems in this study all had sufficient path following accuracy (less than 1cm) that this additional capability was not necessary, but it is an avenue for future research.

It is important to note, however, that this algorithm measures all errors in camera space. The initial target is designated in camera images and the final location is also calculated from camera images (with or without EPEC corrections applied). Even if the target and final poses match perfectly, there could be some bias error between the camera designation an dthe true world frame location. Since the goal of this system is only to move the end-effector to a target designated through vision, any bias error is ignored. Additionally, for most systems where this algorithm would be used, this bias would likely be small.

4. Simulation Environment

To assess the performance of the EPEC algorithm under a wide range of error conditions, a simulation environment where arbitrary errors could be set and the performance of the algorithm measured was created. With the assumption that the closed loop joint control system produces zero steady state error, discrepancies in the final positioning of the arm can be attributed to errors between the actual hardware (arm and cameras) and the associated models. To simulate this, two copies of each model are used: a nominal set to represent the models used in the control software, and a perturbed set with controlled errors to represent real hardware. This simulation environment is illustrated in Figure 2.

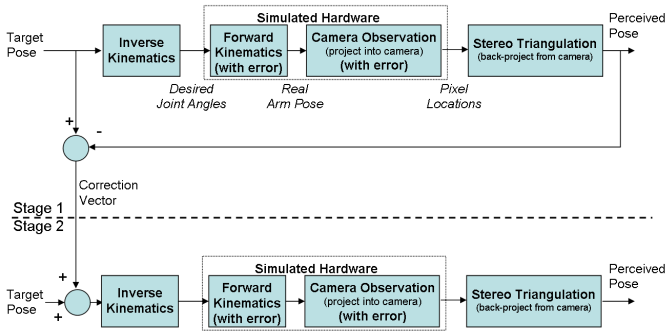

Figure 2. simulation environment block diagram

Given a target pose, inverse kinematics are performed using a nominal arm model, this provides a set of desired joint angles. To simulate the robot moving to these joint angles, forward kinematics is performed on a second arm model which has some amount of error introduced. The result of this process is the actual Cartesian pose of the arm which, in the real system, is not known since there is no ground truth data. The true value of this pose is not known; it can only be observed with the camera system. This observation is another two stage process which involves projecting the 3-D arm location into camera pixel space (a 2D pixel location on each camera CCD). This is done first with a nominal camera model. This camera location is then projected back as one ray from each camera through a second set of camera models with errors introduced. The intersection of these two rays (or its closest approximation) is the perceived arm location. The discrepancy between the commanded pose and the perceived pose is the correction vector.

In this setup, a similar set of forward and inverse operations is performed on both sets of models. For validation, the same camera and arm models can be used in this process and the result will be no final positioning error. The location the arm was observed would be exactly where it was commanded to move.

To asses the performance of EPEC, the process described above is performed in two stages, as illustrated in the figure. In stage 1 of the simulation, the correction vector with a target pose is calculated. This correction vector is then added to a target designated in stage 2 and the process is repeated to calculate the final error. The poses designated in stage 1 and 2 do not need to be the same. The experiments outlined below have a goal of observing the performance degradation as these two poses move further apart.

5. Parameter Space

For the stereo cameras, CAHVOR models [6] were used and calculated with an unsurveyed calibration method [1], which means that the ground truth measurements of arm, camera, and calibration targets is not known. All models produced are relative to the camera frame. Each model has 18 possible parameter values. For simplicity, the radial lens distortion parameters (R) were ignored in simulation and it was assumed that the CCD pointing vector was aligned with the lens vector (O=A). This reduces the number of camera parameters to 12 per camera. Because the cameras are used in a pair to perform stereo triangulation, not the individual parameter values but errors between the two cameras are important. To account for this, two identical camera models were produced with one model was offset by an appropriate baseline of 10cm. Errors were introduced into only one of the two cameras while the other was left fixed.

For the arm kinematics, the standard Denavit-Hartenberg convention [5] was used on the 5 degree of freedom arm. This produces 4 parameters for each link for a total of 20. The arm is arranged in a yaw-pitch-pitch-pitch-yaw configuration. An additional transformation was used to relate the fiducial to the end effector. No errors were introduced into this transform.

With this set of parameters defined, it was possible to examine how errors in each parameter individually affect the total positioning error and how they are compensated with the EPEC algorithm. This makes it possible to identify key parameters

which may require additional consideration during the calibration process. A representative plot of this one parameter analysis is shown in Figure 3. On the left is a plot of correction vectors caused by error in the first yaw angle parameter. All errors radiate perpendicular to the arm direction and grow as the distance from the first yaw joint increases. This is to be expected with a rotational error. The error magnitude from this parameter error is manageable and sufficiently compensated with the EPEC algorithm. The right side of Figure 3 shows positioning errors when compensated with EPEC. Performing the same procedure on the other 31 parameters produced similar results.

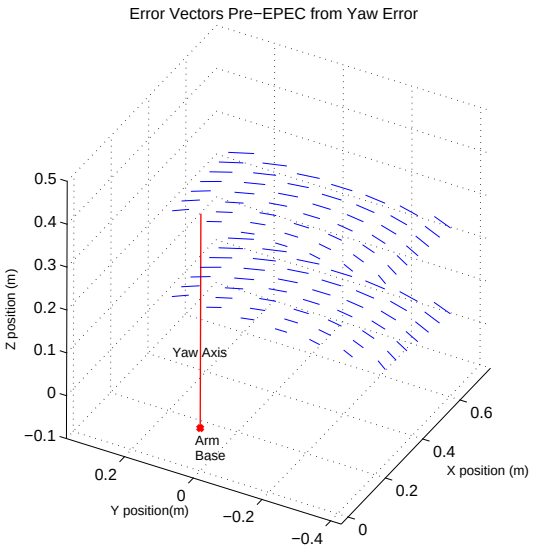

Figure 3. EPEC Correction Vectors plotted in arm workspace; error introduced only to the arm yaw parameter and resulting errors are concentric with arm yaw

To perform a full analysis of this large number of parameters, a statistical approach was employed. Three levels of error were chosen for each of the parameter types (lengths, angles, etc.) and these levels were used as the standard deviation in a random distribution. Three groups of parameter errors were selected: arm only, camera only, and combined. Table 1 outlines these three groups with the standard deviations for their three levels of intensity. This created 9 parameter groups, each with 100 members whose values are a random distribution which adheres to the standard deviations in Table 5.

The intent for this analysis is to observe the combined influence of all parameter errors in a combined manner on the arm motion and EPEC correction. For each of the tests outlined below, all 100 parameter sets in each group are tested and the results are averaged.

6. PERFORMANCE SIMULATIONS

EPEC at the Target

The first goal of this analysis was to determine the effectiveness of the EPEC algorithm. To this effect, this first test was designed to test how the algorithm would perform when an images was available directly at the target location. In this test, the arm was commanded to a target and the initial error was calculated. The error corresponds to the correction vector at that target. To implement EPEC, the arm was commanded again, this time with the correction vector added, and the final positioning error was calculated. This represents the best performance possible with only one iteration.

Table 2 outlines the results of this test on the nine elements of the parameter space.

EPEC Locality

The second goal of this analysis was to determine the effectiveness of a correction vector when it is applied away from the place where it was calculated. To test this, a correction vector was calculated at one location and then applied to targets throughout the workspace. Of interest now is the final positioning error at a target along with the distance between the target and the place where the correction vector was calculated.

For each of the 9 parameter sets, three starting locations were selected to calculate the correction vector. This vector was then applied to each of the targets throughout the workspace. The mean error was calculated at each target and plotted vs. distance from the location where the correction vector was calculated. The result of this test for the first parameter set is shown in Figure 4. The three colors in this plot represent the three targets: target 1 in the center of the arm workspace, target 3 in the center of the camera field of view, and target 2

Table 1. parameter set error magnitudes

Set Name	Length Error	Angle Error	Camera Position	Camera Rotation	Focal Length	Pixel Center
Arm 1	1.0	0.5	X	X	X	X
Arm 2	1.0	1.0	X	X	X	X
Arm 3	2.5	1.0	X	X	X	X
Cam 1	X	X	1.0	0.05	1.0	5.0
Cam 2	X	X	2.5	0.15	2.5	12.5
Cam 3	X	X	5.0	0.30	5.0	25.0
All 1	1.0	0.5	1.0	0.05	1.0	5.0
All 2	1.0	1.0	2.5	0.15	2.5	12.5
All 3	2.5	1.0	5.0	0.30	5.0	25.0
units:	mm	deg.	mm	deg.	pixel	pixel

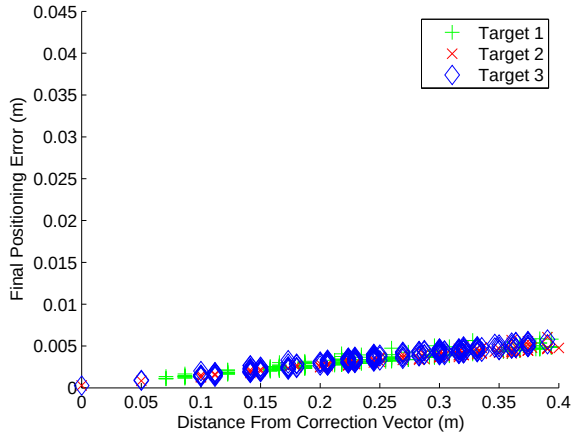

Figure 4. final positioning error vs. distance from correction vector for mild errors in arm DH parameters only (arm set 1)

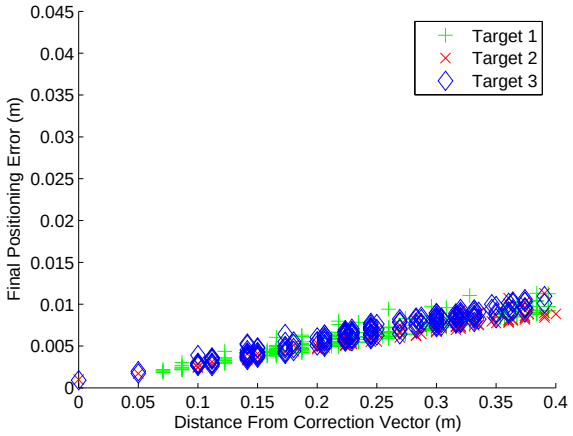

Figure 5. final positioning error vs. distance from correction vector for increased errors in arm DH parameters only (arm set 2)

on the extreme edge of the workspace. In all cases, the three targets agree and it is possible to calculate a trend line.

Figures 4 through 6 illustrate the three cases of arm kinematic errors only. In all three cases, the final positioning error grows linearly with distance from the correction vector location.

The three sets in this group represent three levels of increasing error and likewise the three trend lines on these sets have increasing slope. As errors in the arm model increase, the area where a correction vector can produce a desired level of positioning performance will decrease linearly. This is not surprising, since Denavit-Hartenberg parameters consist of two types: displacement and rotations. Small errors in the link lengths or offsets in the kinematics would correspond to similar small displacements of the end-effector. Assuming a small angle approximation, the same is true for arm angles and link twists.

Errors in the extrinsic camera parameters correspond to displacements and rotations of the camera frame with respect to the world (or arm) frame and would result in the same displacement of the end-effector described above for arm kinematics (assuming again a small angle approximation). Errors in the intrinsic parameters of the camera model, however, will result in ranging errors, which grow quadraticly with distance from the camera. As the chosen intrinsic parameter errors are rather severe, it is not surprising that the overall error increases quadraticly with distance. The results for these camera error cases are shown in Figures 7-9. In these cases, there is a quadratic increase in the positioning error as the correction vector is applied away from where it was calculated. Again, as with the arm error cases, the slope of this trend increases as the severity of the introduced error is increased.

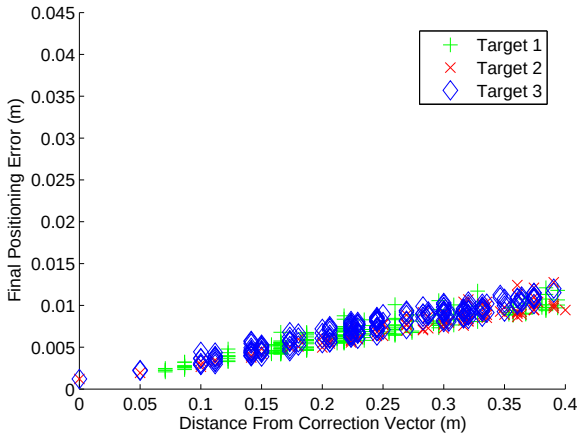

Figure 6. final positioning error vs. distance from correction vector for severe errors in arm DH Parameters only (arm set 3)

Table 2. error magnitudes in mm for EPEC at the target

Set	No EPEC		With EPEC	
Name	Mean Err.	Std. Dev.	Mean Err.	Std. Dev.
Arm 1	12.2	0.7	0.3	0.01
Arm 2	23.0	1.2	0.9	0.05
Arm 3	25.7	1.5	1.2	0.07
Cam 1	3.3	2.0	0.06	0.07
Cam 2	8.4	5.4	0.40	0.45
Cam 3	15.9	9.7	1.50	1.70
All 1	4.9	1.0	0.09	0.04
All 2	10.8	3.2	0.50	0.35
All 3	19.0	9.4	1.90	1.80

Figures 10-12 show the results of the three combined error sets. Not surprisingly, combining the linear and quadratic error trends produces another quadratic trend. Figure 13 shows the trend lines from all 9 parameter groups without the indi-

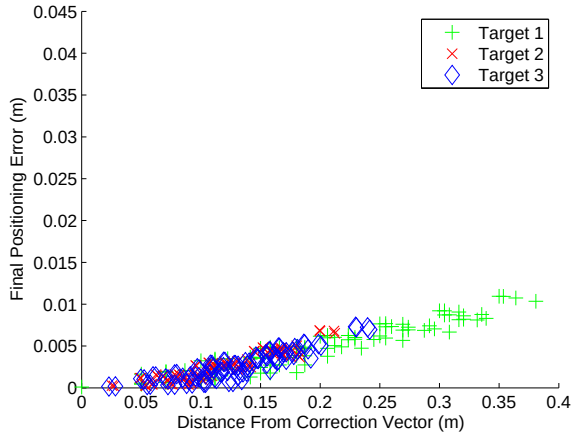

Figure 7. final positioning error vs. distance from correction vector for mild camera errors (camera set 1)

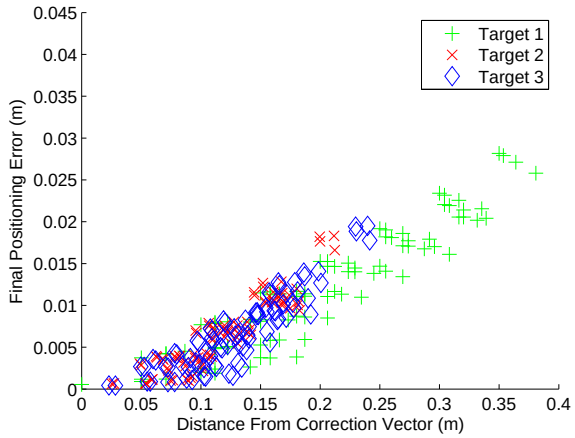

Figure 8. final positioning error vs. distance from correction vector for increased camera errors (camera set 2)

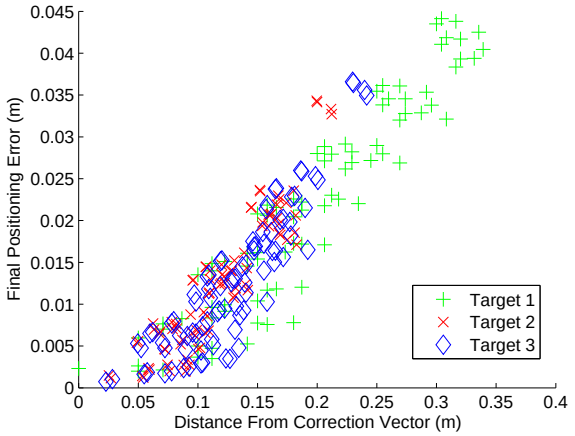

Figure 9. final positioning error vs. distance from correction vector for severe camera errors (camera set 3)

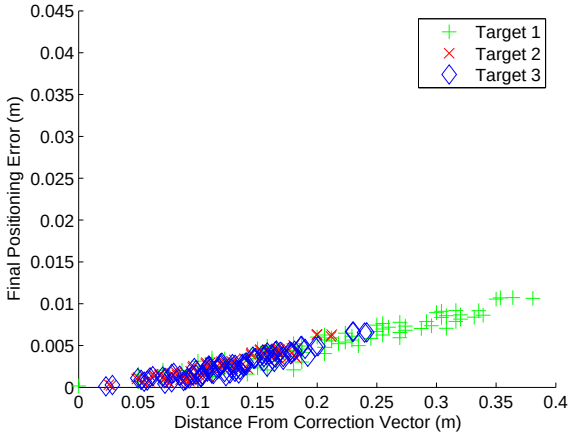

Figure 10. final positioning error vs. distance from correction vector for mild errors in arm and camera (combined set 1)

vidual data points. For the error magnitudes chosen, the camera model errors dominate the algorithm performance. The magnitude of the camera only errors is substantially higher than the kinematic errors. Combining the two produces a locality plot that is only slightly worse than camera errors only.

7. Experimental Results

Experimental Setup

To validate these simulations, EPEC was tested on a representative hardware arm/camera system. The test arm, shown in Figure 14, is a wall mounted 5 degree of freedom manipulator with identical configuration and similar sizing to the MER IDD, approximately 80cm at full extension. The cameras are rigidly mounted to the robot's back plate, have a 10cm baseline, and are angled at 30°down. They have 2.8mm lenses, a 640x480 pixel CCD with 4.65um pixel size and a field of view of about 90°. At 1m away, this configuration corresponds to 1 pixel translating to approximately 1mm in lateral error and a 0.3 pixel error in stereo disparity matching error resulting in 10mm of range error. However, because the algorithm uses a target designated in the same stereo pair that it uses to compute the error vector, it can achieve much better placement performance in practice.

Correction Vector Measurements

The first experiment performed was simply a measurement of correction vectors throughout the workspace. The arm was commanded to 100 target points evenly spaced at 5cm throughout the workspace. At each target, the arm location was calculated through kinematics and through vision and their difference, the correction vector, is plotted in Figure 15. The vectors are anchored (with a dot) at their location calculated by vision and they point toward the commanded target, showing the direction of the correction.

This experiment serves as a baseline for positioning perfor-

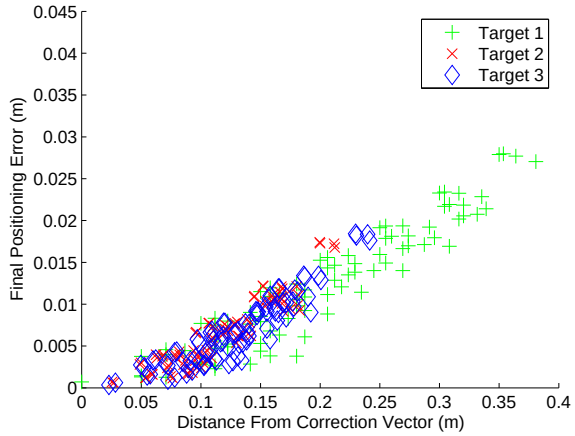

Figure 11. final positioning error vs. distance from correction vector for increased errors in arm and camera (combined set 2)

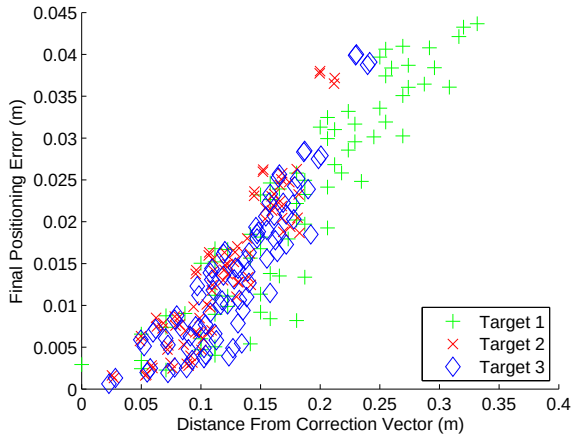

Figure 12. final positioning error vs. distance from correction vector for severe errors in arm and camera (combined set 3)

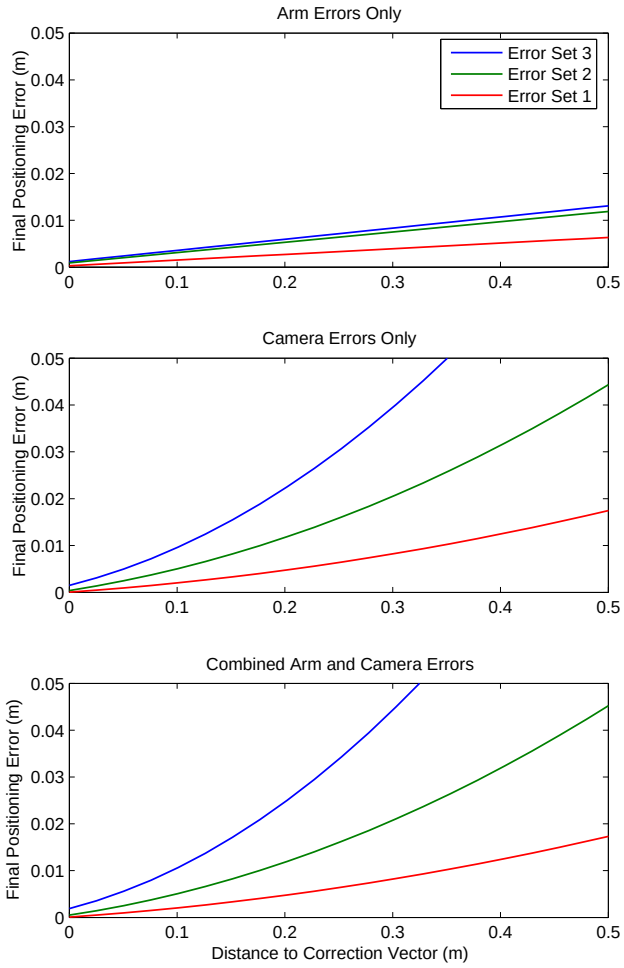

Figure 13. combined best fit trend lines for figures 4-12

mance and also gives a visualization of the modeling error through the workspace. The mean size of these 100 correction vectors is 1.5cm. A comparison of these results to the individual parameter error measurements discussed in Section 5 may reveal particular parameters of concern. For instance, if all error vectors radiate outwards from the camera centers (as these do) there is most likely an error in a parameter that effects stereo ranging, such as the baseline measurement or camera rotation.

Correction Vector Locality

The second experiment performed was an exact copy of the simulated locality experiment described in section Section 6. The arm was commanded to three different start positions and a correction vector is calculated associated with that point. This correction vector is then applied to each of a large number of targets spread throughout the workspace. Using a laser rangefinder total station (Leica Geosystems model TCRA1103+), the target location is measured along with position of the arm before and after the application of the correction vectors. The final positioning error (distance from the target location) is plotted against distance from the three correction vectors. Figure 16 shows the result.

There is a large (+/- 2mm) spread on all the data collected, but this can be attributed to measurement accuracy and human operator error. As expected, there is no relationship between distance to target and positioning when EPEC is not used. The mean of these targets is also identical to the 1.5cm calculated in the previous experiment. With EPEC, however, there is a clear trend in better performance for targets close to the correction vector. An exact fit with noisy data would require more data points, but a simple linear fit gives a good idea of EPEC performance as both a linear and quadratic fit would be very close at the small scales (10cm or less) in question.

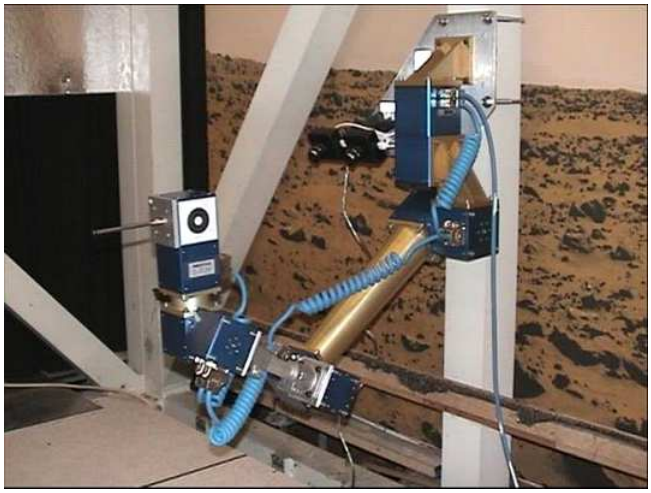

Figure 14. modular arm and camera system used for experiments

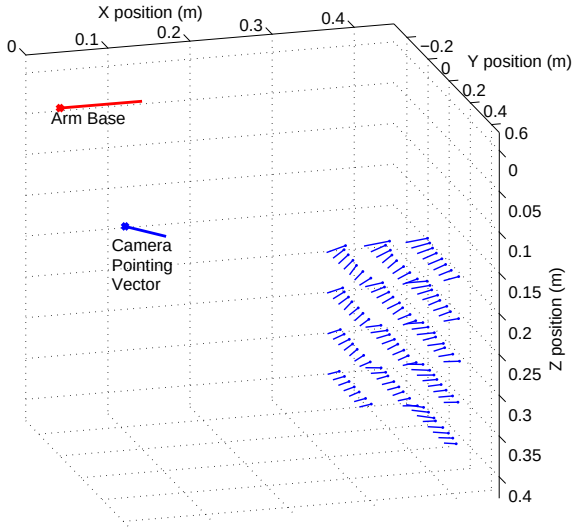

Figure 15. correction vector field for experimental system, plotted in world coordinates with arm base and camera location

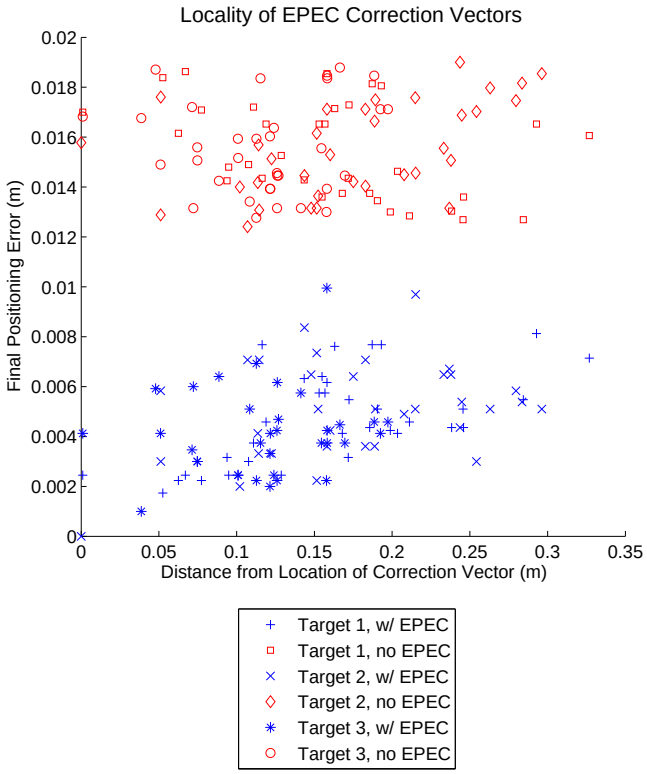

Figure 16. locality experiment performed on modular manipulator, final positioning error with and without EPEC

MER Results

The EPEC algorithm was employed as a ground tool to aid in motion planning on mission sol 698 on the MER rover Opportunity. On the previous sol, three arm commands were uploaded and images with corresponding joint angles were recorded. With this information it was possible to measure the correction vector and upload corrected target locations. On sol 701 the images of the arm at the new, corrected positions were received. Figure 17 shows one set of before and after images and Table 3 outlines the before and after positioning results.

On the first and third attempts there was significant improvement on the order of 1cm to 1mm. However, the second attempt resulted in a strong reflection of the sun off a portion of the end effector and the images were saturated near the fiducial. On this attempt the results were no better than the baseline. Additionally, as there was no ground truth measurement available to verify these numbers so any biases in the vision system (such as fiducial detector bias) would be present in all measurements.

8. Conclusions

The analysis shown here has provided compelling evidence that the EPEC algorithm can make significant improvement in arm positioning accuracy with a simple and computationally efficient implementation. Analysis of the correction vector locality was also critical in ensuring that this algorithm would work under non-ideal conditions. The simulation predicts that final positioning errors will grow smoothly and predictably, a result which was verified through experiment. Given the constraints and hardware capabilities in the class of operations explored in this research, EPEC appears to be an excellent method for improving positioning, either as a standalone algorithm or in conjunction with existing methods.

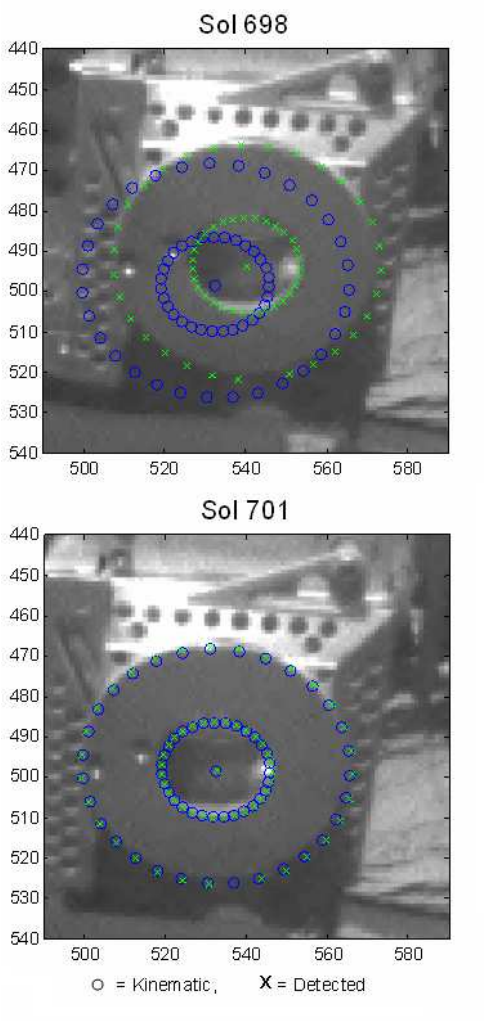

Figure 17. application of EPEC correction on Opportunity IDD, sols 698 and 701

Table 3. positioning error with EPEC on Opportunity IDD, sols 698 and 701

	Error Before Correction	Correction Vector (m)	Error After Correction
Attempt 1	6.9 mm	(-.00499736, -0.00470137, -0.00100508)	0.50 mm
Attempt 2	6.8 mm	(-.00523693, -0.00310702, -0.0030674)	7.2 mm
Attempt 3	11.9 mm	(-.00956444, -0.000652192, -0.00700548)	0.39 mm

9. FUTURE WORK

Comparison Study

Currently underway is a comparison study to gauge the differences in performance and computational requirements between EPEC and two other vision guided manipulation algorithms: Hybrid Image Plane/Stereo (HIPS) and DH-Tune. These three methods each address the issue of errors in camera and arm models differently. HIPS compensates error by recalibrating the camera models only, DH-Tune recalculates the arm models only, while EPEC makes corrections in Cartesian space while keeping both models unchanged.

Fiducial Detection

There are numerous improvements which can be made to the implementation of this algorithm to improve performance. The most immediate of these is the use of sub-pixel fiducial detection. This would improve the camera triangulation accuracy, but may require the use of specific fiducial patterns. Use of a new sub-pixel fiducial detector which uses a circular target and a model based fit is currently being investigated. The use of three fiducials to extract azimuth and elevation information is also under consideration.

MER Integration

An additional avenue of research is the use of EPEC to process archived data. Because the camera images and arm positions are saved for every command sent to the rovers, it is possible to scour the database of over 700 sols of data and calculate correction vectors whenever a proper fiducial (such as the Mossbauer spectrometer ring) is visible. Research is underway to use these data to give a time history of arm/camera degradation or as an offline EPEC correction vector table.

MSL Scale Tests

The robot arm on the proposed Mars Science Laboratory (MSL) mission is more than double the length of the MER IDD. Camera ranging errors and kinematic positioning errors will both be magnified on a longer arm. For this reason it is expected that the baseline positioning performance on an MSL size arm will be more than 1cm. Characterizing this error and observing the improvement caused by the use of the EPEC algorithm is another avenue of continuing research.

10. ACKNOWLEDGEMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration as part of the Mars Technology Program.

REFERENCES

[1] A. Ansar. Survey Free Camera Calibration. JPL Technical Report NTR# 42336, 2005.

[2] P. Backes, A. Diaz-Calderon, M. Robinson, M. Bajracharya, and D. Helmick. Automated Rover Positioning and Instrument Placement. IEEE Aerospace Conference Proceedings, March 2005.

[3] E. T. Baumgartner, et al. The Mars Exploration Rover

Instrument Positioning System. IEEE Aerospace Conference Proceedings, March 2005.

[4] E. T. Baumgartner and P. S. Schenker. Autonomous Image-Plane Robot Control for Martian Lander Operations. Proceedings of the IEEE International Conference on Robotics and Automation, April 1996.

[5] J. Denavit and R. S. Hartenberg. "A kinematic notation for lower-pair mechanisms based on matrices," ASME Journal of Applied Mechanics, vol. 77, no. 2, pp. 215-221, June 1955.

[6] D. Gennery. Least-Squares Camera Calibration Including Lens Distortion and Automatic Editing of Calibration Points. Calibration and Orientation of Cameras in Computer Vision, A. Grun and T. Huang, editors, Springer-Verlag, ISBN 3-540-65283-3, Chapter 5, pp. 123-136, 2001.

[7] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. IEEE Trans. Robotics and Automation, vol. 12, no. 5, pp 651-670, Oct. 1996.

[8] T. Huntsberger, Y. Cheng, A. Stroupe, H. Aghazarian. Closed Loop Control for Autonomous Approach and Placement of Science Instruments by Planetary Rovers. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2005.

[9] J. P. Lewis, "Fast normalized cross-correlation," Vision Interface, 1995.

[10] J. N. Maki, et al. Mars Exploration Rover Engineering Cameras. Journal of Geophysical research, Vol. 108, No. E12, 8071, doci:10.1029/2003JE002077, 2003.

[11] K. Nickels. Hand-Eye Calibration for Robonaut. NASA Summer Faculty Fellowship Program 2003 Final Report, Johnson Space Center.

[12] L. Pedersen, R. Sargent, M. Bualat, M. Deans, C. Kunz, S. Lee, A. Wright. Single Cycle Instrument Deployment. International Symposium on Artificial INtelligence, Robotics and Automation in Space (i-SAIRAS), 2003.

[13] M. Seeling, J.D. Yoder, E. T. Baumgartner and S. B. Skaar. High-Precision Visual Control of Mobile Manipulators. IEEE Transactions on Robotics and Automation, Vol 18, No. 6, December 2002.

[14] S. B. Skaar, W. H. Brockman, and R. Hanson. Camera space manipulation. Int. Journal of Robotics Research, vol. 6, no. 4, pp. 20-32, Winter 1987.

[15] J. Shi, C. Tomasi, "Good Features to Track", IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 1994.

[16] Edward Tunstel, Mark Maimone, Ashitey Trebi-Ollennu, Jeng Yen, Rich Petras, Reg Willson, "Mars Exploration Rover Mobility and Robotic Arm Operational Performance," 2005 IEEE International Conference on Systems, Man, and Cybernetics, October 10-12, 2005, Waikoloa Hilton Village, The Big Island, Hawaii, USA, 2005.

Matthew DiCicco *is an associate member of technical staff at the NASA Jet Propulsion Laboratory. He received a masters degree from MIT where he studied high power robotic manipulators for the Navy. At JPL he has continued his work on robotic manipulators and is currently performing research tasks for the Mars Science Lab rover mission as a member of the Mobility and Manipulation Group.*

Max Bajracharya *is a member of the Mobility and Manipulation Group at the Jet Propulsion Laboratory, Pasadena, CA. His research includes vision-based tracking, manipulation, navigation, and pose estimation for Mars rovers. He is currently the task lead for several tasks focusing on autonomous target approach and instrument placement. Max received his Bachelors and Masters degrees in computer science and electrical engineering from MIT in 2001.*

Kevin M. Nickels *is an associate professor in the Department of Engineering Science at Trinity University. He received the B.S. degree in Computer and Electrical Engineering from Purdue University (1993), and received the M.S. degree (1996) and the Ph. D. (1998) in Electrical Engineering from The University of Illinois at Urbana- Champaign. He is currently working in the areas of computer vision, pattern recognition, and robotics. Dr. Nickels spent the 2005-2006 academic year as a Senior Engineer in Avionics Systems and Space Technology at NASAs Jet Propulsion Laboratory, concentrating on Vision Guided Manipulation.*

Paul Backes *is a technical group leader in the Autonomy and Control section at Jet Propulsion Laboratory, Pasadena, CA, where he has been since 1987. His current activities focus on planetary manipulation, sample acquisition, and sample handling. He received the BSME degree from U.C. Berkeley in 1982, MSME*

in 1984 and Ph.D. in 1987 in Mechanical Engineering from Purdue University. Dr. Backes received the 1993 NASA Exceptional Engineering Achievement Medal for his contributions to space telerobotics, 1993 Space Station Award of Merit, Best Paper Award at the 1994 World Automation Congress, 1995 JPL Technology and Applications Program Exceptional Service Award, 1998 JPL Award for Excellence, 1998 NASA Software of the Year Award Sole Runner-up, and 2004 NASA Software of the Year Award. He has served as an Associate Editor of the IEEE Robotics and Automation Society Magazine.